Probabilistic Programs with Discrete Distributions and Precedence Constrained Knapsack Polyhedra

Andrzej Ruszczyński

RUTCOR, Rutgers Center for Operations Research
Rutgers University, Piscataway, NJ 08854, U.S.A.
rusz@rutcor.rutgers.edu

Abstract

We consider stochastic programming problems with probabilistic constraints involving random variables with discrete distributions. They can be reformulated as large scale mixed integer programming problems with knapsack constraints. Using specific properties of stochastic programming problems and bounds on the probability of the union of events we develop new inequalities for these mixed integer programs. We also develop methods for lifting these inequalities. These procedures are used in a general iterative algorithm for solving probabilistically constrained problems. The results are illustrated with a numerical example.

Keywords: Probabilistic Programming – Integer Programming – Valid Inequalities

1 Introduction

Reliability and risk are key issues in models arising in insurance, finance, telecommunication and many other areas. When incorporated into optimization problems, they take the form of *probabilistic constraints*.

Stochastic programming problems with probabilistic constraints can be introduced as follows. We have a probability space $(\Omega, \mathcal{B}, \mathbb{P})$ and the space $\mathcal{X}$ of measurable mappings $x : \Omega \to \mathbb{R}^n$. Next, we are given a functional $f : \mathcal{X} \to \mathbb{R}$, a measurable constraint function $g : \mathbb{R}^n \times \mathbb{R}^s \to \mathbb{R}^m$, a random vector $\xi : \Omega \to \mathbb{R}^s$, and a set $X \subset \mathcal{X}$. The problem is to find

$$\begin{aligned} \min \quad & f(x) \\ \text{subject to} \quad & \mathbb{P}\{g(x(\omega), \xi(\omega)) \geq 0\} \geq 1 - \alpha, \\ & x \in X, \end{aligned} \tag{1.1}$$

where the symbol $\mathbb{P}$ denotes probability and $\alpha \in (0, 1)$ is some prescribed level.

The simplest case is the *here–and–now* problem in which the decision x is not allowed to depend on the random vector ξ, that is, $X \subseteq \mathbb{R}^n$.

A more involved situation occurs in the *two-stage* case, in which x has two subvectors, $x = (x_1, x_2)$, the first of which has to be determined without the knowledge of the random outcome, while the second one, x_2, can be decided upon *after* $\xi(\omega)$ is known. Then X can contain only decision rules of form $x(\omega) = (x_1, x_2(\omega))$. In a more involved *multistage model* we have $x = (x_1, \dots, x_T)$, where T is the number of stages, and each part x_t of the decision vector may use some partial information available at stage t. The Reader is referred to the book of Birge and Louveaux[3] for an extensive treatment of different information structures in stochastic programming models.

Programming under probabilistic constraints has a long history. Charnes, Cooper and Symonds in [5] formulated probabilistic constraints individually for each stochastic constraint. Joint probabilistic constraints for independent random variables were used first by Miller and Wagner in [11]. The general case was introduced and first studied by Prékopa in [15].

Much is known about problem (1.1) in the case when the decisions x are deterministic vectors in $\mathbb{R}^n$, f is linear in x, and

$$g(x, \xi) = Tx - \xi, \tag{1.2}$$

with some random vector ξ and a deterministic matrix T. In particular, if ξ has a continuous distribution, [18] is an excellent reference. Much less is known in the case of a discrete distribution of ξ (see [6, 19]). When the dependence of g on ξ is more involved, for example the matrix T in (1.2) is random, too, significant difficulties arise. We should mention here the works [9] and [8] on stochastic routing problems, where inequalities eliminating infeasible routes have been developed.

We shall focus our efforts on the case when there are only finitely many realizations $\xi^1, \dots, \xi^N$ of the random vector ξ, occurring with probabilities $p_1, \dots, p_N$. We shall call them *scenarios*. As a result, only finitely many solution realizations $x^i = x(\xi^i)$ may occur, $i = 1, \dots, N$. To facilitate formulation of probabilistic constraints in this case, let us introduce the indicator function $\chi : \mathbb{R}^m \to \{0, 1\}$:

$$\chi(u) = \begin{cases} 1 & \text{if } u \geq 0, \\ 0 & \text{otherwise.} \end{cases}$$

Problem (1.1) can be then written in a more explicit form:

$$\begin{aligned} \min \quad & f(x) \\ \text{subject to} \quad & \sum_{i=1}^{N} p_i \chi(g(x^i, \xi^i)) \geq 1 - \alpha, \\ & x \in X. \end{aligned} \tag{1.3}$$

Let us keep in mind that the set X in the above formulation takes care of the information restrictions on x. For example, in the here-and-now problem, the set X contains only such decisions x that $x^1 = \cdots = x^N$.

Discrete distributions arise frequently in applications, either directly, or as empirical approximations of the underlying distribution $\mathbb{P}$. In the latter case ξ^i are independent observations of ξ, and $p_i = 1/N$ for $i = 1, \dots, N$. If more than one observation have identical outcomes we may still formally treat them as different scenarios.

Throughout, we assume that the functions $f(\cdot)$ and $g(\cdot, \xi^i)$, $i = 1, \dots, N$, are continuous and the set X is compact. Thus, if (1.3) has a nonempty feasible set, an optimal solution exists.

The main observation around which we plan to focus our research is that in many cases one can define a partial order $\preceq$ on the set of scenarios: for some pairs of scenarios i and j we shall be able to say that i is 'not harder' than j. In the case when

$$g(x, \xi) = t(x) - \xi$$

for some function $t : \mathbb{R}^n \to \mathbb{R}^m$ (and with $s = m$) the order $\preceq$ is defined as the componentwise inequality between the right hand side realizations:

$$i \preceq j \;\Leftrightarrow\; \xi^i \le \xi^j.$$

This has been extensively exploited in our recent work with D. Dentcheva and A. Prékopa [6] where we show that only a limited number of scenarios play a role in the problem. These are $(1-\alpha)$-*efficient points* v^j defined as the minimal points (in the sense of the partial order $\le$) of the set of realizations ξ^i for which

$$\mathbb{P}\{\xi \le \xi^i\} \ge 1 - \alpha.$$

In [6] we developed an algorithm that iteratively updates the set of relevant $(1-\alpha)$-efficient points to generate tight lower and upper bounds for probabilistically constrained problems.

In section 2 we introduce a more general definition of a *consistent* order and we show that it can be defined for many classes of probabilistically constrained problems. This will be exploited in section 3 to formulate deterministic equivalents of probabilistically constrained problems with the use of *precedence constrained knapsack polyhedra.* We shall discuss valid inequalities for probabilistic constraints, obtained from *induced covers* and we shall formulate auxiliary problems to find valid inequalities of interest. Section 4 is devoted to specialized lifting procedures for these inequalities. In section 5 we shall construct a method for solving probabilistically constrained problems that uses valid inequalities developed in the preceding sections. Finally, in section 6 we shall have a numerical illustration.

We shall use the symbol $\preceq$ to denote a partial order relation in a set I; the strict relation $i \prec j$ will be understood in a usual way ($i \preceq j$ and $i \neq j$). The sets of minimal and maximal elements of I under the order $\preceq$ will be denoted $\mathcal{M}(I)$ and $\mathcal{S}(I)$, respectively.

2 Consistent orders of scenarios

We start from the definition of an 'easier' scenario.

Definition 2.1. *A partial order $\preceq$ on $\{1, \dots, N\}$ is* consistent *with problem (1.3) if for every $x \in X$ there exists $\bar{x} \in X$ such that*

(i) $f(\bar{x}) \leq f(x)$;

(ii) $\sum_{i=1}^{N} p_i \chi(g(\bar{x}^i, \xi^i)) \geq \sum_{i=1}^{N} p_i \chi(g(x^i, \xi^i))$; *and*

(iii) *for all* $i, j \in \{1, \dots, N\}$ *one has*

$$(i \prec j) \ \wedge \ (g(\bar{x}^j, \xi^j) \geq 0) \Rightarrow (g(\bar{x}^i, \xi^i) \geq 0).$$

The order $\preceq$ *is* strongly consistent *if condition (iii) holds for* $\bar{x} = x$.

Let us consider two practically important cases of probabilistically constrained stochastic programming problems when a consistent order can easily be defined.

We start from the linear problem with joint probabilistic constraints:

$$\begin{aligned} \min \quad & c^T x \\ \text{subject to} \quad & \sum_{i=1}^{N} p_i \chi(T^i x - h^i) \geq 1 - \alpha, \\ & x \in X, \end{aligned} \tag{2.1}$$

with scenarios $i = 1, \dots, N$ characterized by realizations (T^i, h^i) of an $m \times n$ random matrix T and a random vector $h \in \mathbb{R}^m$. The convex closed polyhedron $X \subseteq \mathbb{R}^n$, the cost vector $c \in \mathbb{R}^n$ and the probability level $\alpha \in (0, 1)$ are given. From Definition 2.1 we obtain the following result.

Lemma 2.2. *The partial order* $\preceq$ *defined on* $\{1, \dots, N\}$ *as follows*

$$i \preceq j \ \Leftrightarrow \ h^i - T^i x \leq h^j - T^j x \text{ for all } x \in X$$

is strongly consistent with problem (2.1).

In a special case, if $X = \mathbb{R}^n_+$ we have

$$i \preceq j \ \Leftrightarrow \ T^i \geq T^j \text{ and } h^i \leq h^j.$$

When only the right hand side h is random, the order $\preceq$ is identical to the component-wise inequality $\leq$ in the space of realizations of h, whose implications for our problem are thoroughly analyzed in [6].

Let us now define the linear two-stage problem with probabilistic constraints. It has two groups of decision variables: first stage decisions $x \in \mathbb{R}^n$ and second stage decisions $y^i \in \mathbb{R}^l$ associated with each scenario $i = 1, \dots, N$. The problem is formulated as follows:

$$\begin{aligned} \min \quad & c^T x + \sum_{i=1}^{N} p_i \langle q, y^i \rangle \\ \text{subject to} \quad & \sum_{i=1}^{N} p_i \chi(T^i x + W y^i - h^i) \geq 1 - \alpha, \\ & x \in X, \\ & y^i \in Y, \quad i = 1, \dots, N. \end{aligned} \tag{2.2}$$

In addition to the notation explained at (2.1), $Y \subseteq \mathbb{R}^l$ is a convex closed polyhedron, and $q^i \in \mathbb{R}^l$ is a given second stage cost vector associated with scenario $i = 1, \dots, N$. The probabilities of scenarios are denoted $p_1, \dots, p_N$.

Lemma 2.3. *The partial order $\preceq$ defined on $\{1, \dots, N\}$ as follows*

$$(i \preceq j) \;\Leftrightarrow\; (p_i = p_j) \;\wedge\; (h^i - T^i x \le h^j - T^j x) \;\forall\, x \in X$$

is consistent with problem (2.2).

Proof. Let $\hat{x}$ and $\hat{y}^i$, $i = 1, \dots, N$, be an optimal solution of (2.2). Consider two scenarios, i and j, such that $i \prec j$. Suppose that $T^j \hat{x} + W\hat{y}^j \ge h^j$ but $T^i \hat{x} + W\hat{y}^i \not\ge h^i$. Define a new the second stage solution $\widetilde{y}$ by switching in $\hat{y}$ the values of y^i and y^j. By the definition of $\preceq$, the point $(\hat{x}, \widetilde{y})$ is feasible for (2.2) and the objective value at it is no greater than at $(\hat{x}, \hat{y})$. Consequently, it is optimal for (2.2). By carrying out the above transformation finitely many times we can construct an optimal solution $(\hat{x}, \bar{y})$ at which the order $\preceq$ satisfies Definition 2.1. □

3 Mixed integer formulation and induced covers

Let us reformulate problem (1.3) as a mixed integer program. To this end we find for each $i = 1, \dots, N$ a vector $d^i \in \mathbb{R}^m$ such that

$$g(x^i, \xi^i) + d^i \ge 0, \quad \text{for all} \quad x \in X.$$

Such a vector exists, because $g(\cdot, \xi^i)$ is continuous and X compact.

This allows us to transform (1.3) to a mixed integer program:

$$\min \quad f(x) \tag{3.1}$$

$$\text{subject to} \quad g(x^i, \xi^i) + d^i z_i \ge 0, \quad i = 1, \dots, N, \tag{3.2}$$

$$\sum_{i=1}^{N} p_i z_i \le \alpha, \tag{3.3}$$

$$x \in X, \tag{3.4}$$

$$z_i \in \{0, 1\}, \quad i = 1, \dots, N. \tag{3.5}$$

If f is convex and $g(\cdot, \xi^i)$ concave for all ξ^i, the above problem is a mixed integer convex program; its relaxation (with the integrality restriction (3.5) ignored) can be efficiently solved by convex programming methods. However, the full mixed integer program appears to be very difficult, since the number of scenarios N may be very large. To reduce its complexity we shall use the partial order $\preceq$ associated with (1.3). From Definition 2.1 we obtain the following observation.

Lemma 3.1. *If $\preceq$ is a consistent order for (3.1)–(3.5), then there exists an optimal solution $(\hat{x}, \hat{z})$ of (3.1)–(3.5) such that for all $i, j \in \{1, \dots, N\}$*

$$(i \preceq j) \Rightarrow (z_i \le z_j).$$

Therefore, adding to (3.1)–(3.5) the constraints

$$z_i \leq z_j \quad \text{for all } i, j \in \{1, \dots, N\} \text{ such that } i \preceq j \tag{3.6}$$

does not cut off *all* optimal solutions.

Inequalities (3.3) and (3.6), together with the integrality restriction (3.5), define a *precedence constrained knapsack polyhedron* (PCKP), extensively studied in combinatorial optimization [4, 14, 10]. We shall adapt and develop some of the ideas introduced for PCKPs in order to gain more insight into problem (3.1)–(3.6) and to create efficient methods for its solution.

Let us define the sets

$$A_i = \{j \in \{1, \dots, N\} : i \preceq j\}, \quad i = 1, \dots, N.$$

Germane to our research is the concept of the *induced cover*, which generalizes the classical notion of a cover for knapsack constraints (see [12, 20] and the references therein).

Definition 3.2. *A set $C \subseteq \{1, \dots, N\}$ is called an* induced cover *if*

$$\mathbb{P}\Big\{ \bigcup_{i \in C} A_i \Big\} > \alpha. \tag{3.7}$$

An induced cover C is proper*, if for every $j \in C$*

$$\mathbb{P}\Big\{ \bigcup_{i \in C \setminus \{j\}} A_i \Big\} \leq \alpha \tag{3.8}$$

and minimal *if for every $j \in C$*

$$\mathbb{P}\Big\{ \bigcup_{i \in C} A_i \setminus \{j\} \Big\} \leq \alpha. \tag{3.9}$$

For any induced cover C we have a valid inequality:

$$\sum_{i \in C} z_i \leq |C| - 1. \tag{3.10}$$

Indeed, if $z_i = 1$ for all $i \in C$ then (3.6) and the definition of A_i imply that $z_k = 1$ for $k \in \bigcup_{i \in C} A_i$. Consequently, (3.7) contradicts (3.3).

The notion of a minimal induced cover for PCKP has been introduced in [4] and analyzed in [10, 14]. Van de Leensel, van Hoesel and van de Klundert prove in [10] that inequalities (3.10) generated by minimal induced covers are facet defining for subsets of PCKP and they use the general lifting algorithm of Balas [1] to obtain facet defining inequalities for the entire PCKP.

In the context of probabilistic programming, though, the application of these results encounters difficulties due to the large number N of possible scenarios. The enumeration of all proper induced covers is practically impossible. Lifting of the covers, as shown in

[10], requires the solution of very many knapsack subproblems, each of them NP-hard. We shall therefore concentrate on two issues: finding *relevant* proper induced covers and determining their effective lifting. Our main objective is to incorporate these techniques into a specialized method for solving probabilistically constrained problems of form (1.3).

The first question we are interested in is the following: given a set $I \subseteq \{1, \dots, N\}$ and a fractional point $\tilde{z} \in [0, 1]^N$ find an induced cover $C \subseteq I$ such that the inequality (3.10) cuts-off $\tilde{z}$, that is,

$$\sum_{i \in C} \tilde{z}_i > |C| - 1. \tag{3.11}$$

Of course, the only interesting case is with I being an induced cover itself. To find the deepest cut (3.11) we introduce binary variables v_i, $i \in I$, to decide whether scenario i will be included in C or not, and we formulate the optimization problem:

$$\min \quad \sum_{i \in I} (1 - \tilde{z}_i) v_i \tag{3.12}$$

$$\text{subject to} \quad \mathbb{P}\Big\{ \bigcup_{i\,:\,v_i = 1} A_i \Big\} > \alpha, \tag{3.13}$$

$$v_i \in \{0, 1\}, \quad i \in I. \tag{3.14}$$

Frm Definition 3.2 we deduce the following result.

Lemma 3.3. *Assume that I is an induced cover. If the optimal value of (3.12)–(3.14) is smaller than 1, the set $C = \{i \in I : v_i = 1\}$ defines an induced cover for which inequality (3.11) is satisfied. If the optimal value is greater or equal than 1, there is no induced cover $C \subseteq I$ such that inequality (3.11) holds.*

Problem (3.12)–(3.14) is still a difficult combinatorial optimization problem, especially due to the implicit constraint (3.13). We shall derive a restriction of this problem in a form of a linear program. Let us introduce additional decision variables y_{ij}, $i, j \in I$, $i < j$, and modify problem (3.12)–(3.14) as follows:

$$\min \quad \sum_{i \in I} (1 - \tilde{z}_i) v_i \tag{3.15}$$

$$\text{subject to} \quad \sum_{i \in I} v_i \mathbb{P}\{A_i\} - \sum_{\substack{i,j \in I \\ i<j}} y_{ij} \mathbb{P}\{A_i \cap A_j\} \geq \alpha + \epsilon, \tag{3.16}$$

$$y_{ij} \geq v_i + v_j - 1, \quad y_{ij} \geq 0, \quad i, j \in I, \quad i < j, \tag{3.17}$$

$$v_i \in \{0, 1\}, \quad i \in I, \tag{3.18}$$

with $0 < \epsilon < \min_{1 \leq i \leq N} p_i$.

Lemma 3.4. *If problem (3.15)–(3.18) has a solution, the set $C = \{i \in I : v_i = 1\}$ is an induced cover. Moreover, if the optimal value is smaller than 1, then inequality (3.11) is satisfied.*

Proof. Let $(\hat{v}, \hat{y})$ be the optimal solution of (3.15)–(3.18). With no loss of feasibility we may assume that $\hat{y}_{ij} = \hat{v}_i \wedge \hat{v}_j$. Then (3.16) takes on the form

$$\sum_{i\in C} \mathbb{P}\{A_i\} - \sum_{\substack{i,j\in C\\ i<j}} \mathbb{P}\{A_i \cap A_j\} \geq \alpha + \epsilon.$$

Recalling the Boole–Bonferroni inequality (see, e.g., [16])

$$\mathbb{P}\Big\{\bigcup_{i\in C} A_i\Big\} \geq \sum_{i\in C} \mathbb{P}\{A_i\} - \sum_{\substack{i,j\in C\\ i<j}} \mathbb{P}\{A_i \cap A_j\}, \tag{3.19}$$

we conclude that (3.7) holds, that is, C is an induced cover. By assumption, the value of (3.15) is smaller than 1, so $\sum_{i\in C}(1 - \tilde{z}_i) < 1$ which is identical to (3.11). □

The Boole–Bonferroni inequality is not sharp, but problem (3.15)–(3.18) can be refined by clustering the sets A_i.

Definition 3.5. *A collection $J_k \subseteq I$, $k \in K$ is called a* proper partition *of I, if*

(i) $\bigcup_{k\in K} J_k = I$;

(ii) $B_k = \bigcap_{i\in J_k} A_i \neq \emptyset$, $k \in K$; *and*

(iii) $B_k \cap A_i = \emptyset$, *for all* $k \in K$ *and* $i \notin J_k$.

Let $k(i)$ be such that $i \in J_{k(i)}$ for all $i \in I$.

Lemma 3.6. *If J_k, $k \in K$, is a proper partition of I, then*

$$\begin{aligned}\mathbb{P}\Big\{\bigcup_{i\in I} A_i\Big\} \geq{}& \sum_{k\in K} \mathbb{P}\{B_k\} + \sum_{i\in I}(\mathbb{P}\{A_i\} - \mathbb{P}\{B_{k(i)}\})\\ &- \sum_{k\in K} \sum_{\substack{i,j\in J_k\\ i<j}} \Big(\mathbb{P}\{A_i \cap A_j\} - \mathbb{P}\{B_k\}\Big) - \sum_{\substack{i,j\in I, i<j\\ k(i)\neq k(j)}} \mathbb{P}\{A_i \cap A_j\}.\end{aligned}$$

Proof. We have

$$\bigcup_{i\in I} A_i = \bigcup_{k\in K} B_k \cup \bigcup_{i\in I} (A_i \setminus B_{k(i)}).$$

Applying the Boole–Bonferroni inequality to the union on the right hand side and noting that Definition 3.5(iii) implies

$$\mathbb{P}\{(A_i \setminus B_{k(i)}) \cap (A_j \setminus B_{k(j)})\} = \begin{cases} \mathbb{P}\{A_i \cap A_j\} - \mathbb{P}\{B_{k(i)}\} & \text{if } k(i) = k(j),\\ \mathbb{P}\{A_i \cap A_j\} & \text{if } k(i) \neq k(j),\end{cases}$$

we obtain the required result. □

We shall use Lemma 3.6 to refine problem (3.15)–(3.18). Let us denote for brevity, $\mu_i = \mathbb{P}\{A_i\}$, $\mu_{ij} = \mathbb{P}\{A_i \cap A_j\}$, $\rho_k = \mathbb{P}\{B_k\}$ and consider the linear program

$$\min \quad \sum_{i \in I} (1 - \tilde{z}_i) v_i \tag{3.20}$$

$$\text{subject to} \quad \sum_{k \in K} \rho_k \lambda_k + \sum_{i \in I} v_i (\mu_i - \rho_{k(i)})$$

$$- \sum_{k \in K} \sum_{\substack{i,j \in J_k \\ i<j}} y_{ij} \Big(\mu_{ij} - \rho_k \Big) - \sum_{\substack{i,j \in I, i<j \\ k(i) \neq k(j)}} y_{ij} \mu_{ij} \ \geq \ \alpha + \epsilon, \tag{3.21}$$

$$y_{ij} \geq v_i + v_j - 1, \quad y_{ij} \geq 0, \quad i,j \in I, \quad i<j, \tag{3.22}$$

$$\lambda_k \leq \sum_{i \in J_k} v_i, \quad \lambda_k \leq 1, \quad k \in K, \tag{3.23}$$

$$v_i \in \{0,1\}, \quad i \in I. \tag{3.24}$$

Proposition 3.7. *If problem (3.20)–(3.23) has a solution, then the set $C = \{i \in I : v_i = 1\}$ is an induced cover. Moreover, if the optimal value is smaller than one, then inequality (3.11) is satisfied.*

Proof. Let us observe that with no loss of feasibility we may set $y_{ij} = v_i \wedge v_j$ and $\lambda_k = \bigvee_{i \in J_k} v_i$. Define

$$\tilde{J}_k = J_k \cap C, \qquad \tilde{K} = \{k \in K : \tilde{J}_k \neq \emptyset\}.$$

The sets $\tilde{J}_k$, $k \in \tilde{K}$, define a proper partition of C. Using Lemma 3.6 and the inclusion

$$\tilde{B}_k = \bigcap_{i \in \tilde{J}_k} A_i \supseteq B_k, \quad k \in \tilde{K},$$

we conclude that (3.21) implies (3.7). The remaining part of the proof is identical with the proof of Lemma 3.4. ☐

Inequality (3.21) is stronger than (3.16) by the quantity

$$\sum_{k \in K} |J_k|(|J_k| - 2)\mathbb{P}\{B_k\}.$$

The sets B_k can be found by the following greedy algorithm: J_1 is the largest set of scenarios belonging to the largest number of sets A_i; after deleting $i \in J_1$ we define J_2 in the same way, etc.

4 Lifting

Let us now consider the issue of lifting a cover inequality (see [1, 12]). We are not necessarily interested in the optimal lifting, which is known to be a very difficult problem, but rather in a lifting that can be accomplished relatively easy, by linear programming.

Suppose that we have an induced β-cover: a set C such that

$$\sum_{i\in C} z_i \le \beta \tag{4.1}$$

is a valid inequality, where $\beta \le |C|-1$. For a scenario $s \notin C$ we want find (γ_s, β_s) such that the inequality

$$\sum_{i\in C} z_i + \gamma_s z_s \le \beta_s \tag{4.2}$$

is valid for the PCKP.

Let us first consider the case when

$$s \notin \bigcup_{i\in C} A_i.$$

We shall search for a lifting in a form of a β-cover inequality, assuming $\beta_s = \beta$ and checking whether we can set $\gamma_s = 1$ in (4.2). This can be decided by solving the following combinatorial problem

$$\max \quad \sum_{i\in C} v_i \tag{4.3}$$

$$\text{subject to} \quad \mathbb{P}\Big\{A_s \cup \bigcup_{i\,:\,v_i=1} A_i\Big\} \le \alpha, \tag{4.4}$$

$$v_i \in \{0,1\}, \quad i \in C. \tag{4.5}$$

If the optimal value of this problem is smaller than β we can set $\gamma_s = 1$; otherwise $\gamma_s = 0$ (lifting is unsuccessful). After that, we can process the next candidate variable, etc.

Problem (4.3)–(4.5) is a difficult combinatorial optimization problem. It was considered in [10] (with a different notation) and proved to be NP-hard. In our setting, in view of a very large number of scenarios, solving it in its pure form appears to be very difficult, especially because it has to be carried out for every candidate variable to be included in the valid inequality.

We shall develop relaxations of problem (4.3)–(4.5) which will be easier to solve and which will generate valid liftings, although (possibly) missing some lifting opportunities. To this end we shall adapt and modify the probability bounding approach based on binomial moments developed in [17].

For random events A_i, $i \in I$, we define p_m to be the probability that exactly m out of $n = |I|$ events happen. The probabilities p_m, $m = 1, \dots, n$, satisfy the binomial moment equations

$$\sum_{m=r}^{n} \binom{m}{r} p_m = \sum_{i_1<i_2<\cdots<i_r} \mathbb{P}\{A_{i_1} \cap A_{i_2} \cap \cdots \cap A_{i_r}\}, \quad r = 1, \dots, n. \tag{4.6}$$

The probability that at least one of these events happens equals

$$\mathbb{P}\Big\{\bigcup_{i\in I} A_i\Big\} = \sum_{m=1}^{n} p_m. \tag{4.7}$$

Using these relations, we construct the following linear programming problem

$$\max \quad \sum_{i\in I} v_i \tag{4.8}$$

$$\text{subject to} \quad \sum_{m=1}^{n} p_m \le \alpha, \tag{4.9}$$

$$\sum_{m=1}^{n} m p_m = \sum_{i\in I} v_i \mathbb{P}\{A_i\}, \tag{4.10}$$

$$\sum_{m=2}^{n} \binom{m}{2} p_m = \sum_{i<j} y_{ij} \mathbb{P}\{A_i \cap A_j\}, \tag{4.11}$$

$$y_{ij} \ge v_i + v_j - 1, \quad 0 \le y_{ij} \le \min(v_i, v_j), \quad i, j \in I, \quad i < j, \tag{4.12}$$

$$v_i \in \{0, 1\}, \quad i \in I. \tag{4.13}$$

$$p_m \ge 0, \quad m = 1, \dots, n. \tag{4.14}$$

Proposition 4.1. *Let $\bar{\beta}$ be the optimal value of problem (4.8)–(4.14). Then $\sum_{i\in I} z_i \le \bar{\beta}$ is a valid inequality.*

Proof. Suppose that the assertion is not true. Then there exists a set $J \subseteq I$ of cardinality $|J| > \bar{\beta}$ such that

$$\mathbb{P}\Big\{\bigcup_{i\in J} A_i\Big\} \le \alpha.$$

Define $v_i = 1$ if $i \in J$, and $y_{ij} = v_i \wedge v_j$. Also, let p_m be the probability that exactly m events out of the collection A_i, $i \in J$, happen. Then (4.6)–(4.7) imply that the constraints (4.9)–(4.11) are satisfied. The other constraints (4.12)–(4.14) are satisfied by construction. Thus $|J| = \sum_{i\in I} v_i \le \bar{\beta}$, a contradiction. □

To lift the cover C in (4.1) we apply the above result with $I = C \cup \{s\}$ and we enforce $v_s = 1$ (we already have a valid inequality without s). If the optimal value $\bar{\beta}$ does not exceed β, we can add z_s to the inequality; that is, replace C with $C \cup \{s\}$ in (4.1).

In (4.10)–(4.11) we use only two first binomial moment constraints, rather than all of them, and therefore constraint (4.9) is a relaxation of (4.4). We could have included higher order binomial moment constraints to improve the quality of this relaxation, but in the context of stochastic programming it would be highly unrealistic, due to the large number of combinations of events A_i to be considered. Instead of that, we shall try to refine problem (4.8)–(4.14) by using the information that is readily available.

First, it is easy to calculate for each A_i the probability

$$\delta_i = \mathbb{P}\Big\{ A_i \setminus \bigcup_{j \in I \setminus \{i\}} A_j \Big\}.$$

Then we must have $p_1 \geq \sum_{i \in I} \delta_i v_i$; inequality is needed here because C is a subset of I.

Second, a substantial refinement can be gained by employing clustering. Let, again J_k, $k \in K$, be a proper partition of I. As before, we denote $\mu_i = \mathbb{P}\{A_i\}$, $\mu_{ij} = \mathbb{P}\{A_i \cap A_j\}$, $\rho_k = \mathbb{P}\{B_k\}$. Consider the problem

$$\max \quad \sum_{i \in I} v_i \tag{4.15}$$

$$\text{subject to} \quad \sum_{m=1}^{n} p_m + \sum_{k \in K} \rho_k \lambda_k \leq \alpha \tag{4.16}$$

$$\sum_{m=1}^{n} m p_m = \sum_{i \in I} (\mu_i - \rho_{k(i)}) v_i, \tag{4.17}$$

$$\sum_{m=2}^{n} \binom{m}{2} p_m = \sum_{k \in K} \sum_{\substack{i,j \in J_k \\ i<j}} y_{ij} \Big(\mu_{ij} - \rho_k \Big) + \sum_{\substack{i,j \in I, i<j \\ k(i) \neq k(j)}} y_{ij} \mu_{ij}, \tag{4.18}$$

$$y_{ij} \geq v_i + v_j - 1, \quad 0 \leq y_{ij} \leq \min(v_i, v_j), \quad i, j \in I, \quad i < j, \tag{4.19}$$

$$\lambda_k \leq \sum_{i \in I} v_i, \quad 0 \leq \lambda_k \leq 1, \quad k \in K, \tag{4.20}$$

$$v_i \in \{0, 1\}, \quad i \in I, \tag{4.21}$$

$$p_1 \geq \sum_{i \in I} \delta_i v_i, \tag{4.22}$$

$$p_m \geq 0, \quad m = 2, \ldots, n. \tag{4.23}$$

Similarly to Proposition 4.1, using the observations from the proof of Proposition 3.7 we obtain the following result.

Proposition 4.2. *Let $\bar{\beta}$ be the optimal value of problem (4.15)–(4.23). Then $\sum_{i \in I} z_i \leq \bar{\beta}$ is a valid inequality.*

Problem (4.15)–(4.23), although it appears rather involved, is much easier to solve than the 'compact' formulation (4.3)–(4.5), because standard linear programming methods can be employed.

Let us now consider lifting with respect to scenarios

$$s \in \bigcup_{i \in C} A_i. \tag{4.24}$$

The case when C is a minimal induced cover is well studied in [10] and the ideas employed there are readily applicable to our problem. To illustrate them in our context, we can formulate the following result.

Lemma 4.3. *Let C be an induced cover, J_k, $k \in K$, be a proper partition of C, and let $j_k \in \bigcap_{i\in J_k} A_i$. Then the inequality*

$$\sum_{i\in C} z_i + \sum_{k\in K}(|J_k|-1)(1-z_{j_k}) \le |C|-1 \tag{4.25}$$

is a valid inequality for the PCKP.

Proof. The assertion follows from the observation that $z_{j_k} = 0$ implies $z_i = 0$ for all $i \in J_k$. □

Unfortunately, the practical relevance of the cover inequalities lifted with respect to the scenarios s satisfying (4.24) is rather limited. Indeed, consider the continuous relaxation of problem (3.1)–(3.6) (obtained by ignoring (3.5)) and suppose that $(\tilde{x}, \tilde{z})$ is its optimal solution. Define $V = \{i : \tilde{z}_i > 0\}$. Clearly, we need valid inequalities only if $\sum_{i\in V} p_i > \alpha$; otherwise the current solution is optimal for (3.1)–(3.6).

Let $C \subset V$ be an induced cover satisfying the assumptions of lemma 4.3. If the lifted inequality (4.25) can be satisfied by setting $z_{j_k} = 1$ for all clusters k, we shall obtain a new optimal solution of the relaxed problem. At this solution, the values of decision variables x, the set V and the objective value are exactly the same as before. On the other hand, if making $z_{j_k} = 1$ does not restore feasibility, the same effect can be obtained from the basic cover inequality (3.10), to which (4.25) reduces in this case.

For these reasons we shall not explore the lifting with respect to scenarios satisfying (4.24).

5 Cut and branch method for probabilistic constraints

Let us now turn to ways of solving the mixed integer formulation (3.1)–(3.6) with the application of valid inequalities developed in sections 3 and 4. Define the sets

$$\begin{aligned}
&S_0 = \{z \in \mathbb{R}^N : \sum_{i=1}^{N} p_i z_i \le \alpha,\ z_i \le z_j \text{ for all } i,j \in \{1,\dots,N\} \text{ such that } i \preceq j\},\\
&B_0 = \{z \in \mathbb{R}^N : 0 \le z_i \le 1,\ i = 1,\dots,N\},\\
&L_0 \subseteq \{1,\dots,N\}.
\end{aligned}$$

We shall construct sequences of sets S_k, B_k and L_k, $k = 1,2,\dots$, by adding valid inequalities to the definition of S_k, fixing to $\{0,1\}$ some variables in B_k, and selecting subsets of relevant scenarios to be included into L_k.

Step 0 Set $k = 0$.

Step 1 Solve the relaxed problem

$$\begin{aligned}
\min\quad & f(x) && (5.1)\\
\text{subject to}\quad & g(x^i,\xi^i) + d^i z_i \ge 0, \quad i \in L_k, && (5.2)\\
& x \in X, && (5.3)\\
& z \in S_k \cap B_k. && (5.4)
\end{aligned}$$

Let $(\hat{x}^k, \hat{z}^k)$ denote the solution found, with scenario solutios $(\hat{x}^{ki}, \hat{z}_i^k)$, $i = 1, \dots, N$.

Step 2 Define the sets

$$H_k = \{i \in \{1, \dots, N\} : g(\hat{x}^{ki}, \xi^i) \geq 0\},$$
$$I_k = \{1, \dots, N\} \setminus H_k.$$

If $\sum_{i \in I_k} p_i \leq \alpha$ then stop; otherwise continue.

Step 3 Find an induced cover $C_k \subseteq \mathcal{M}(I_k)$ (recall that $\mathcal{M}(I_k)$ is the set of minimal elements in I_k).

Step 4 For each $s \in \mathcal{M}(I_k) \setminus C_k$ lift the cover C_k to obtain a $|C_k|$-cover $\hat{C}_k \subseteq \mathcal{M}(I_k)$.

Step 5 Set

$$S_{k+1} = S_k \cap \Big\{ z \in \mathbb{R}^N : \sum_{i \in \hat{C}_k} z_i \leq |C_k| - 1 \Big\}.$$

Step 6 If $\mathcal{M}(I_k) \subseteq L_k$ and $\hat{z}^k \in S_{k+1}$, choose $b_k \in \mathcal{M}(I_k)$ such that $z_{b_k}^k \in (0,1)$ and set $B_{k+1} = \{z \in B_k : z_{b_k} \in \{0,1\}\}$; otherwise set $B_{k+1} = B_k$.

Step 7 Choose $L_{k+1} \supseteq L_k \cup \mathcal{M}(I_k)$ increase k by one and go to Step 1.

Theorem 5.1. *After finitely many iterations the algorithm stops at a point $(\hat{x}^k, \hat{z}^k)$ such that $\hat{x}^k$ is optimal for (1.3).*

Proof. Let us show that if the algorithm does not stop at iteration k, Steps 3–6 can be executed. Since $\sum_{i \in I_k} p_i > \alpha$, the set $\mathcal{M}(I_k)$ is an induced cover, so Step 3 can be carried out. The induced cover C_k is a legitimate outcome of Step 4, too. Step 5 defines a nonempty set S_{k+1}, because it always contains 0. It remains to analyze Step 6.

Suppose that $\mathcal{M}(I_k) \subseteq L_k$. By (5.2), $\hat{z}_i^k > 0$ for all $i \in \mathcal{M}(I_k)$. Then, by the definition of S_0, $\hat{z}_i^k > 0$ for all $i \in I_k$. If a fractional component $\hat{z}_{b_k}^k$ cannot be found, we must have $\hat{z}_i^k = 1$ for all $i \in I_k$. But then $\hat{z}^k$ violates the cover inequality $\sum_{i \in \hat{C}_k} z_i \leq |C_k| - 1$, so $\hat{z}^k \notin S_{k+1}$. Consequently, if $\hat{z}^k \in S_{k+1}$, a fractional coordinate $\hat{z}_{b_k}^k$ exists.

The above argument shows that the algorithm is well defined. If it does not stop, then $S_{k+1} \subseteq S_k$, $B_{k+1} \subseteq B_k$, and $L_{k+1} \supseteq L_k$, and at least one of these inclusions is strict.

There are finitely many covers possible, so finitely many different sets S_k may occur. The number of possible sets B_k and L_k is finite, too. Therefore, the algorithm must stop at Step 2 at some iteration k^*.

Problem (5.1)–(5.4) is a relaxation of (3.1)–(3.6). By setting $z_i = 1$ if $\hat{z}_i^{k^*} > 0$, and $z_i = 0$ otherwise, we can satisfy all constraints of (3.1)–(3.6) without changing the objective value. Therefore the solution $\hat{x}^{k^*}$ is optimal for (1.3). □

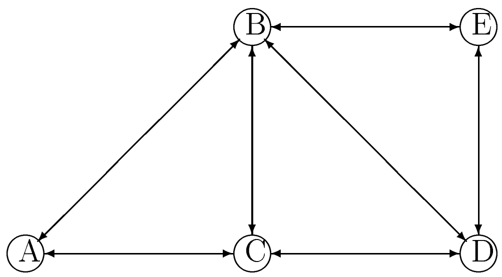

Figure 6.1: The graph of the stochastic multicommodity network flow example.

6 Numerical illustration

Let us consider a stochastic multicommodity network flow problem with the node set $\mathcal{V}$ and arc set $\mathcal{A} \subset \mathcal{V} \times \mathcal{V}$. For each pair of nodes $(k, l) \in \mathcal{V} \times \mathcal{V}$ there is a random quantity d_{kl} to be shipped from k to l. Our objective is to find arc capacities $x(a)$, $a \in \mathcal{A}$, such that the network can carry the flows with a sufficiently large probability $1 - \alpha$ and the capacity expansion cost $\langle c, x \rangle$ is minimized.

Denote the demand scenarios by d^i_{kl}, $i = 1, \dots, N$, and their probabilities by p_i. Introducing the variables $y^i_{kl}(a)$ to denote the flow from k to l passing arc a in scenario i, we can formulate the problem as follows:

$$\min \quad \sum_{a \in \mathcal{A}} c(a)x(a) \tag{6.1}$$

$$\text{subject to} \quad \sum_{a \in \mathcal{A}^+(\nu)} y^i_{kl}(a) - \sum_{a \in \mathcal{A}^-(\nu)} y^i_{kl}(a) = \begin{cases} -d^i_{kl} & \text{if } \nu = k \\ d^i_{kl} & \text{if } \nu = l \\ 0 & \text{otherwise,} \end{cases} \tag{6.2}$$

$$\nu, k, l \in \mathcal{V}, \quad i = 1, \dots, N,$$

$$\sum_{i=1}^{N} p_i \, \chi\Big(x - \sum_{k,l \in \mathcal{V}} y^i_{kl}\Big) \geq 1 - \alpha, \tag{6.3}$$

$$x \geq 0, \quad y \geq 0. \tag{6.4}$$

In the flow balance equations (6.2) we use $\mathcal{A}^-(\nu)$ and $\mathcal{A}^+(\nu)$ to denote the sets of arcs going out of node ν and coming into node ν, respectively.

As an illustration, consider the network shown in Figure 6.1. We assume that the demand is symmetric, that is, $d_{kl} = d_{lk}$ for all pairs (k, l). For $k < l$ we set:

$$d_{kl} = 0.1D + \xi_{kl},$$

where D (the total traffic) has a normal distribution with the expected value 30 and standard deviation 5, and ξ_{kl} are independent normal variables with zero expectation and standard deviation 0.25.

From	To	Cost
A	B	310
A	C	230
B	C	250
B	4	180
B	E	350
C	D	400
D	E	270

Table 6.1: Expansion Costs

The expansion costs are symmetric, too. Table 6.1 gives their values for $k < l$.

Two versions of the problem have been solved: with 100 and with 200 scenarios. In both cases we set $\alpha = 0.1$. These problems are not easy from the point of view of mixed integer programming; for example, the 200 scenario version has 28000 continuous variables, 200 binary variables, and 20001 constraints. They are already too difficult for the standard MIP solver CPLEX. We have to admit here that the choice if the number of scenarios incorporated into the model was fairly arbitrary here. The statistical analysis of the approximation error involved is far beyong the scope of this paper.

We have implemented the cut and branch method of Section 5 in the modeling language AMPL [7]. CPLEX was used as the MIP solver for the master problem at Step 1. It had much fewer binary variables than the full formulation, and could be solved rather effectively.

Figure 6.2 shows the master objective value in successive iterations for both cases. In Figure 6.3 we give the probability that the demand cannot be carried by the capacities equal to the current master's solution. Finally, Figure 6.4 shows the number of variables that are restricted to be binary at the current master's solution.

We see that the method converges rapidly in this example, and the number of binary variables remains moderate. This is due to the fact that the method tries to identify the key scenarios which are located on the boundary of the set of manageable demand realizations. It is worth mentioning that our lifting procedure generated 8 successful liftings in the 100 scenario example, and 10 successful liftings in the 200 scenario example.

The solutions obtained are similar, as can be seen from Table 6.2 (by symmetry, we give only the capacities $x(i, j)$ for $i < j$).

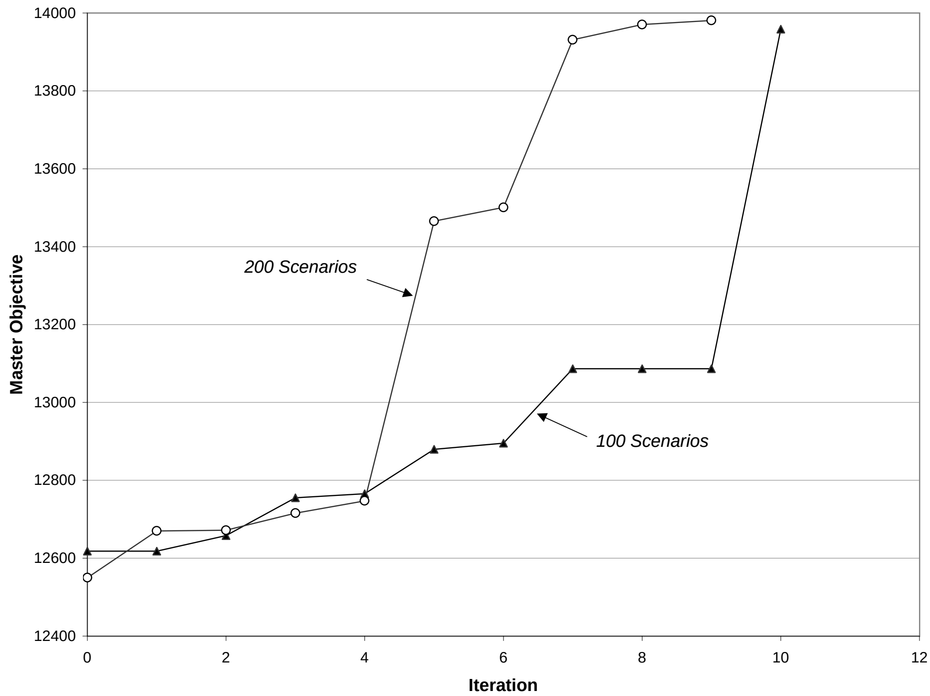

Figure 6.2: The objective value of the master problem.

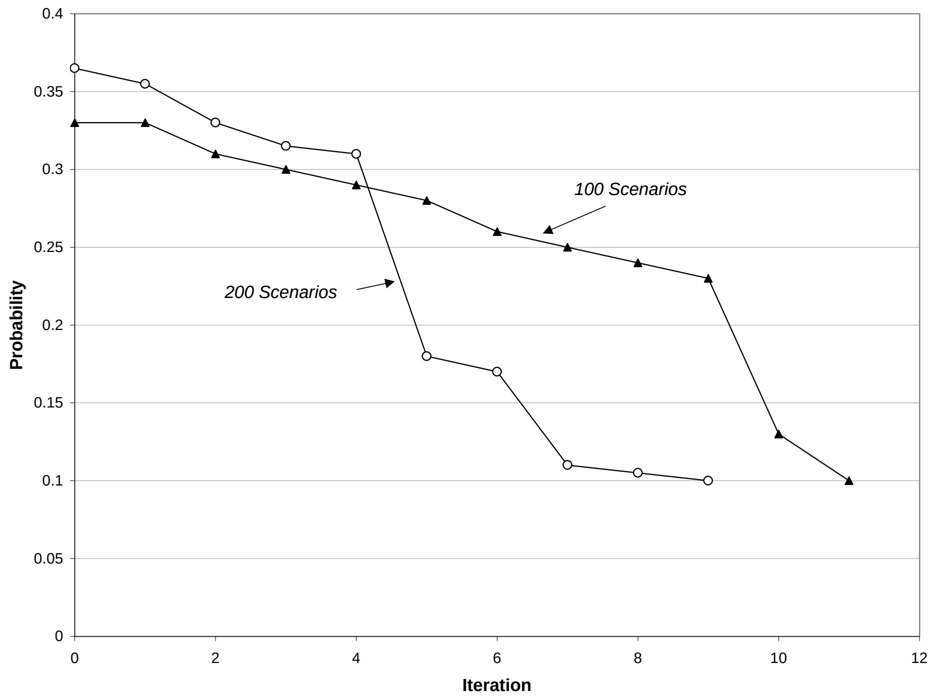

Figure 6.3: The probability that no feasible flow exists.

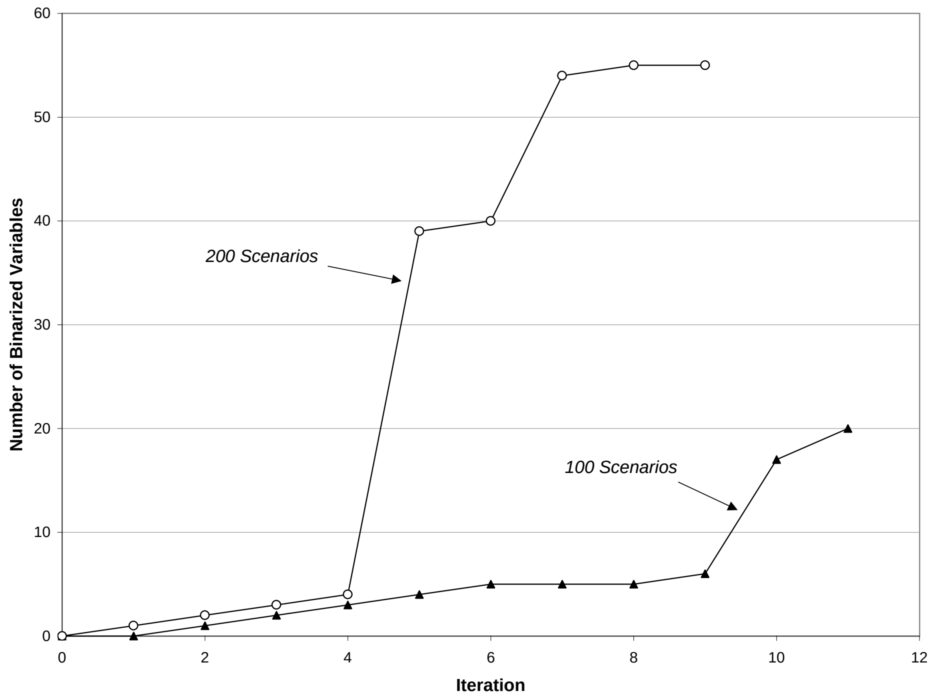

Figure 6.4: The number of variables which are restricted to be binary.

From	To	Capacity	
		100 Scenarios	200 Scenarios
1	2	11.25	10.86
1	3	3.63	3.89
2	3	7.37	7.11
2	4	7.54	7.22
2	5	11.08	10.53
3	4	3.76	3.97
4	5	3.84	4.35

Table 6.2: Optimal arc capacities

References

[1] E. Balas, Facets of the knapsack polytope, *Mathematical Programming* 8 (1975) 146–164.

[2] C. Barnhart, E.L. Johnson, G.L. Nemhauser, M.V.P. Savelsbergh and P.H. Vance, Branch-and-Price: Column Generation for Solving Huge Integer Programs, *Operations Research* 46 (1998) 316–329.

[3] J.R. Birge and F. Louveaux, *Introduction to Stochastic Programming*, Springer-Verlag, New York, NY, 1997.

[4] E.A. Boyd, Polyhedral results for the precedence constrained knapsack problem, *Discrete Applied Mathematics* 41(1993) 185–201.

[5] A. Charnes, W.W. Cooper and G.H. Symonds, Cost Horizons and Certainty Equivalents: An Approach to Stochastic Programming of Heating Oil. *Management Science* 4 (1958) 235–263.

[6] D. Dentcheva, A. Prékopa and A. Ruszczyński, Concavity and efficient points of discrete distributions in probabilistic programming, *Mathematical Programming*, accepted for publication.

[7] R. Fourer, D.M. Gay and B.W. Kernighan, *AMPL: A Modeling Language For Mathematical Programming*,

[8] M. Gendreau, G. Laporte and R. Séguin, Stochastic Vehicle Routing, *European Journal of Operational Research* 88 (1996) 3–12.

[9] G. Laporte, F.V. Louveaux and H. Mercure, Models and Exact Solutions for a Class of Stochastic Location–Routing Problems, *European Journal of Operational Research* 39 (1989) 71–78.

[10] R.L.M.J. van de Leensel, C.P.M. van Hoesel and J.J. van de Klundert, Lifting valid inequalities for the precedence constrained knapsack problem, *Mathematical Programming* 86 (1999) 161–185.

[11] L.B. Miller and H. Wagner, Chance-Constrained Programming with Joint Constraints. *Operations Research* 13 (1965) 930–945.

[12] G.L. Nemhauser and L.A. Wolsey, *Integer and Combinatorial Optimization*, John Wiley & Sons, New York, 1988.

[13] V.I. Norkin, Yu.M. Ermoliev and A. Ruszczyński, On optimal allocation of indivisibles under uncertainty, Operations Research 46 (1998) 381–395.

[14] K. Park and S. Park, Lifting cover inequalities for the precedence constrained knapsack problem, *Discrete Applied Mathematics* 72(1997) 219–241.

[15] A. Prékopa, On Probabilistic Constrained Programming. *Proceedings of the Princeton Symposium on Mathematical Programming.* Princeton University Press, 1970, pp. 113–138.

[16] A. Prékopa, Boole-Bonferroni Inequalities and Linear Programming. *Operations Research* 36 (1988) 145–162.

[17] A. Prékopa, Sharp Bounds on Probabilities Using Linear Programming. *Operations Research* 38 (1990) 227-239.

[18] A. Prékopa, *Stochastic Programming*, Kluwer, Dordrecht, Boston, 1995.

[19] S. Sen, Relaxations for the Probabilistically Constrained Programs with Discrete Random Variables, *Operations Research Letters* 11 (1992) 81–86.

[20] L.A. Wolsey, *Integer Programming*, John Wiley & Sons, New York, 1998.